

Usando Linux en el laboratorio de numérico

Propaganda: Linux es un sistema operativo con grandes cualidades técnicas y que además ha conseguido destruir barreras económicas en la difusión de la informática: ahora cualquier usuario con paciencia y una conexión de red puede disponer, sin ningún desembolso, de una versión en pequeño de los sistemas UNIX (el “hermano mayor” de Linux) usados en los centros de investigación, y una cantidad ingente de *software*.

Es justo mencionar que desde el punto de vista del usuario medio, se puede citar también algunas desventajas: requiere tiempo y conocimientos explotar todas las ventajas de Linux; no se ha desarrollado *software* de ocio (juegos) del nivel de otros sistemas y, sobre todo, puede ser muy complejo configurar adecuadamente dispositivos de *hardware* (*drivers* de impresoras, tarjetas aceleradoras, cámaras digitales) en gran medida porque los fabricantes suelen orientar sus productos al sistema operativo más extendido.

Consola: En las distribuciones Linux modernas casi todo se puede hacer en modo gráfico, sin embargo es muy conveniente conocer los comandos básicos. Pinchando en el icono de la barra de herramientas similar a una pizarra con una concha, se puede abrir una consola con una línea de comandos en la que se escribirán las instrucciones que deseemos.

Matlab: Para abrir Matlab basta teclear `matlab6` en una consola.

Editores: Hay varios editores disponibles. Se aconseja emplear Kedit (kwrite es en principio más avanzado pero la versión del laboratorio es antigua y no tiene configurada la impresión). Se invoca tecleando `kedit nombrefichero`.

Uso de disquetes: Para copiar por ejemplo un fichero llamado `fichero.m` al disquete, escríbase `mcopy fichero.m "a:"`. Para proceder en el otro sentido, del disquete al directorio actual `mcopy "a:fichero.m" ./` las comillas son opcionales en la mayor parte de las situaciones. Con `mdir "a:"` se puede ver el contenido del disquete.

Se recomienda el uso de estos comandos incluso a los que sepan hacer estas operaciones en modo gráfico. Si aun así se emplea, no olvidar la siguiente nota:

MUY IMPORTANTE: Si se ha montado la disquetera (el triángulo verde a la derecha del icono del disquete está encendido), se debe desmontar antes de salir de la cuenta (pinchar en el icono con el botón derecho y seleccionar “desmontar”).

Visores PDF y PS: Un fichero PDF se puede visualizar `xpdf fichero.pdf` o `gv fichero.pdf`. Con `gv` también se visualizan ficheros postscript (con extensiones `.ps` o `.eps`).

Generalidades: El símbolo “&” sirve para enviar un proceso a segundo plano. Esto significa que por ejemplo `matlab6 &` abrirá Matlab dejando la línea de comandos libre.

El tabulador permite completar comandos. Por ejemplo, si en el directorio en curso hay un fichero llamado `nombremuylargo.m` entonces para editarlo, en vez de escribir

`kedit` y el nombre, basta con `kedit n` y pulsar el tabulador. En caso de ambigüedad se mostrarán todas las posibilidades.

El directorio en curso se indica con un punto “.” y el directorio en que está incluido con dos puntos “..”, de ahí la sintaxis citada anteriormente para `mcopy`; tecleando `mcopy ../fichero.m` “a:” se copiaría al disquete `fichero.m` situado en el directorio padre del que está en curso.

Comandos básicos: Ésta es una lista muy breve de algunos de los comandos más usados. Precediendo su nombre con `man` se pueden conocer todas sus opciones y su potencial.

Comando	Significado	Ejemplo
<code>ls</code>	lista de ficheros	<code>ls f*.m</code>
<code>cd</code>	cambio de directorio	<code>cd ./subdirectorio</code>
<code>more</code>	muestra contenido	<code>more fichero1.m</code>
<code>cp</code>	copia	<code>cp fichero1.m ./subdir/fichero2.m</code>
<code>rm</code>	borra fichero	<code>rm fichero1.m</code>
<code>mv</code>	mueve fichero	<code>mv fichero1.m fichrenombrado.m</code>
<code>mkdir</code>	crea directorio	<code>mkdir subdirectorio</code>
<code>rmdir</code>	borra directorio vacío	<code>rmdir subdirectorio</code>
<code>mkdir</code>	crea directorio	<code>mkdir subdirectorio</code>
<code>man</code>	ayuda	<code>man ls</code>
<code>ps -A</code>	lista de procesos	<code>ps -A</code>
<code>killall -9</code>	detiene un proceso	<code>killall -9 matlab6</code>

Nota: Los ficheros borrados no se pueden recuperar. Los conocedores de Linux sabrán que incluyendo una línea `alias rm='rm -i'` en el fichero `.bashrc`, al menos pregunta antes de borrar.

Algunos comandos y ejemplos Matlab

Nos limitaremos aquí a indicar los comandos de más uso en este curso y algunos ejemplos para rápida referencia. De ninguna manera las siguientes líneas son un manual autocontenido de Matlab.

Gráficos

<code>plot</code>	Dibujo de curvas o nubes de puntos en \mathbb{R}^2
<code>plot3</code>	Dibujo de curvas en \mathbb{R}^3
<code>surf</code>	Dibujo de superficies
<code>mesh</code>	Dibujo de superficies

Ejemplos:

```
.....  
% Dibujo de una gráfica simple  
x=linspace(0,13,100); % La x varía entre 0 y 13 con 100 nodos  
y=sin(x)-cos(1.5*x); % La función es y=y(x)  
plot(x,y); % Dibuja la gráfica  
grid; % Dibuja una malla  
xlabel('Eje x'); % Nombre de los ejes  
ylabel('Eje y');  
title('Gráfica de la función'); % Título de la gráfica  
.....
```

```
% Dibujo de una curva en R^3  
t=linspace(0,30,100); % t entre 0 y 100  
plot3(cos(t).*exp(-t/15),sin(t).*exp(-t/15),-t); % Dibuja  
title('Torbellino'); % Título  
axis square % Ejes iguales  
.....
```

```
% Gráfica del paraboloides hiperbólico  
[x y]=meshgrid(linspace(-1,1,15),linspace(-1,1,15)); % Crea una malla  
z=x.^2-y.^2; % Función  
figure(1)  
mesh(x,y,z); % Primero con mesh  
title('mesh: Elementos sin rellenar');  
figure(2)  
surf(x,y,z); % Luego con surf  
title('surf: Elementos rellenos');  
.....
```

Funciones

Funciones elementales:

abs	Módulo, valor absoluto
acos	Arco coseno
asin	Arco seno
atan	Arco tangente
conj	Conjugado
cos	Coseno
exp	Exponencial

floor	Parte entera
log	Logaritmo neperiano
round	Entero más cercano
sin	Seno
sqrt	Raíz cuadrada
tan	Tangente

Funciones definidas por el usuario:

Deben estar en un fichero “.m” cuyo nombre coincida con el de la función. La primera línea que no sea de comentario debe responder al esquema: `function arg. de salida=nombre(arg. de entrada)`.

Ejemplo:

```
.....
% Contenido del fichero acirc.m
function a=acirc(r)
a=pi*r*r; % Área del círculo de radio r
.....
```

Al escribir en la línea de comandos de Matlab `acirc(1)` se obtendrá 3.1416.

Otras funciones y comandos relacionados:

<code>eval</code>	Evalúa la expresión dada por una cadena de caracteres
<code>feval</code>	Evalúa la función dada por una cadena de caracteres
<code>nargin</code>	Número de argumentos de entrada
<code>nargout</code>	Número de argumentos de salida

Por ejemplo, `eval('1+2')` calcula $1 + 2$ y `feval('acirc',1)` calcula `acirc(1)`.

Vectores y matrices

Matlab está dirigido especialmente a cálculo matricial (Mat-lab = Matrix laboratory) por ello gran parte de las funciones indicadas se aplican igualmente a números y a matrices. Indicamos aquí algunas específicas en la creación y manipulación de vectores y matrices.

Función	Descripción	Ejemplo de uso
<code>'</code>	Traspuesta	<code>[1 2; 3 4]'</code> es <code>[1 3; 2 4]'</code>
<code>det</code>	Determinante	<code>det([1 2; 3 4])</code> es -2
<code>diag</code>	Matrix diagonal	<code>diag([1 2 1])</code> tiene en la diagonal 1, 2, 1
<code>diff</code>	Vector de incrementos	<code>diff([1 2 7])</code> es <code>[1 5]</code>
<code>eye</code>	Matrix “unidad”	<code>eye(3)</code> es I_3
<code>length</code>	Longitud de un vector	<code>length([1 2 4])</code> es 3
<code>linspace</code>	progresión aritmética	<code>linspace(1,3,5)</code> es 1, 1.5, 2, 2.5, 3
<code>max</code>	Máximo elemento	<code>max([1 7 4])</code> es 7
<code>meshgrid</code>	Malla (para gráficos)	<code>meshgrid([0 1],[2 3])</code> es <code>[0 1; 0 1]</code>
<code>min</code>	Mínimo elemento	<code>min([1 7 4])</code> es 1
<code>ones</code>	Matriz de unos	<code>ones(2)</code> es <code>[1 1; 1 1]</code>
<code>rand</code>	Matriz aleatoria (unif.)	<code>rand(2,3)</code>
<code>randn</code>	Matriz aleatoria (norm.)	<code>randn(2,3)</code>
<code>rank</code>	Rango	<code>rank([1 2; 3 6])</code> es 1
<code>size</code>	Dimensión	<code>size([1 2 3; 3 3 6])</code> es <code>[2 3]</code>
<code>sparse</code>	Matriz dispersa	<code>sparse([2],[1],[7],6,6)</code> sólo $a_{21} = 7$
<code>sum</code>	Suma filas	<code>sum([1 2; 3 8])</code> es <code>[4 10]</code>
<code>zeros</code>	Matriz de ceros	<code>zeros(1,4)</code> es <code>[0 0 0 0]</code>

Programación

Los comandos básicos son:

<code>break</code>	Sale de un bucle
<code>if...elseif...else</code>	Ejecuta un bloque condicionalmente
<code>end</code>	Fin de un bucle <code>for</code> , <code>while</code> , <code>if</code>
<code>for</code>	Ejecuta un bloque mientras una variable cambia
<code>return</code>	Vuelve al punto de llamada
<code>while</code>	Ejecuta un bloque mientras se cumple una condición

Ejemplos:

```
.....  
  
% Lista de números impares  
for impares=1:2:11 % Del uno al once de dos en dos  
    disp(impares); % Escribe  
end  
  
.....  
  
% Lista de números pares  
n=0;  
while n<12 % Si n es menor que 12  
    n=n+2; % Avanza al siguiente par  
    disp(n); % Escribe  
end  
  
.....  
  
% Dado: 5 gana, 1 pierde  
n=0;  
while n~=5 % Mientras no sea 5  
    n=floor(1+6*rand(1)); % Número aleatorio entre 1 y 6  
    disp('Tirada:');  
    disp(n); % Escribe  
    if n==1  
        disp('Has perdido') % Escribe  
        return  
    end  
end  
disp('Has ganado') % Escribe  
  
.....
```

Entrada y salida de datos

Algunos de los comandos más comunes para la entrada y salida de datos son:

disp	Muestra un texto o variable
fclose	Cierra un fichero
fopen	Abre un fichero
fprintf	Escribe en un fichero
fscanf	Lee de un fichero
ginput	Toma puntos de una figura con el ratón
gtext	Escribe donde se pincha con el ratón
input	Espera una variable
load	Carga variables
menu	Menú de opciones
save	Guarda variables

Ejemplos:

```
.....
x=linspace(0,1,20);
y=x.^2;
plot(x,y); % Dibuja y=x^2
gtext('Has pinchado aquí') % La primera vez que se pincha, escribe texto
[x y]=ginput(3); % Las tres siguientes toma puntos y los guarda en (x,y)
disp([x y]); % Muestra coordenadas
```

```
.....
A=full(sparse([1 1 3 3 3],[1 2 1 2 4],[-1 -2 -3 -4 -5],3,4));
save nada.txt A -ascii % Guarda la matriz A en nada.txt
B = load('nada.txt'); % La recupera con nombre B
```

```
.....
n=input('Introduce un número '); % Recoge un dato
op=menu('Y una opción','opción 1','opción 2','opción 3') % Menú
disp('El número y la opción son');
disp([n op]);
```

```
.....
% Tabla de raíces cuadradas
x=[1:30];
x=sqrt(x)
fid = fopen('nada.txt', 'wt'); % w= escribir t= texto
fprintf(fid, '%f \n', x); % Escribe en fichero
fclose(fid); % Cierra
fid = fopen('nada.txt', 'rt'); % r= leer t= texto
y=fscanf(fid,'%f'); % Lee fichero y lo carga en y
fclose(fid); % Cierra. Ahora x=y'
.....
```